

**Bakalářská práce**  
**Interaktivní výukový kurz**  
**Microsoft Silverlight**  
**Interactive Tutorial**  
**Microsoft Silverlight**

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

## Zadání bakalářské práce

Student:

**Pavel Ferdian**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Interaktivní výukový kurz - Microsoft Silverlight**  
**Interactive Tutorial - Microsoft Silverlight**

Zásady pro vypracování:

Technologie bohatých internetových aplikací je dnes efektivním způsobem tvorby webových aplikací. Cílem této práce je vytvoření výukových ukázek využití platformy MS Silverlight a jejich integrace do výukového kurzu.

1. Stručně charakterizujte platformu MS Silverlight.
2. Navrhněte několik ukázkových aplikací, které zahrnete do výukového kurzu na silverlight.cs.vsb.cz.
3. Aplikace budou sloužit nejen jako podklad pro kurz, ale také jako jeho funkční část (kvíz, videopřehrávač).
4. Do kurzu doplňte multimediální obsah.
5. Zhodnoťte využití technologie Silverlight a jeho výuku formou vytvořeného kurzu.

Seznam doporučené odborné literatury:

M. MacDonald: Pro Silverlight 4 in C#, 2010, Apress, ISBN: 9781430229797

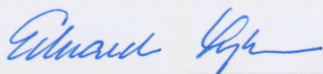
MSDN Library: <http://msdn.microsoft.com/en-us/library/cc645050%28VS.95%29.aspx>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Michal Radecký**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



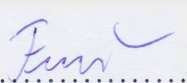
doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

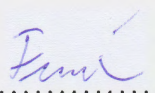
V Ostravě 24. dubna 2012



.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 24. dubna 2012



.....

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Ing. Michalu Radeckému, Ph.D. za cenné rady, připomínky a metodické vedení práce. Dále bych chtěl poděkovat Stanislavu Velkému a Martinu Gluči za ochotu testování aplikací.

## **Abstrakt**

Cílem této práce bylo popsat platformu Silverlight a ukázat základní vlastnosti a možnosti této platformy. Hlavním obsahem práce bylo vytvořit rozšíření existujícího kurzu platformy Silverlight o další kapitoly a interaktivní část, která obsahuje sadu screencastů a tři ukázkové aplikace. Kurz a interaktivní části vytvořili celek, sloužící jako výukový materiál. Dalším obsahem práce bylo popsat postup tvorby kapitol kurzu, screencastů a ukázkových aplikací.

**Klíčová slova:** Silverlight, Screencast, XAML, C#, .NET framework, bakalářská práce

## **Abstract**

The purpose of this study was description of the Silverlight platform and show the basic features and capabilities of this platform. The main content of this study was creation of platform extension for existing courses of Silverlight by adding another chapters and an interactive section, which contains a set of screencasts and three sample applications. This course and the interactive parts created attaching material. Another part of the study was description of the creation process chapter of the course, the screencasts and the demo applications.

**Keywords:** Silverlight, Screencast, XAML, C#, .NET framework, bachelor thesis

## **Seznam použitých zkratk a symbolů**

RIA	– Rich Internet application
WCF	– Windows Communication Foundation
WPF	– Windows Presentation Foundation
WF	– Windows Workflow Foundation
XML	– Extensible Markup Language
HTML	– Hyper Text Markup Language
LINQ	– Language Integrated Query
WP7	– Windows Phone 7
VŠB	– Vysoká škola báňská - Technická univerzita Ostrava

## Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Výukový modul</b>	<b>6</b>
2.1	Cíle . . . . .	6
2.2	Tvorba kapitol . . . . .	6
2.3	Tvorba screencastů . . . . .	7
2.4	Tvorba ukázkových aplikací . . . . .	8
<b>3</b>	<b>Platforma Microsoft Silverlight</b>	<b>9</b>
3.1	Základní vlastnosti platformy Silverlight . . . . .	10
3.2	Jazyk XAML . . . . .	11
3.3	Základní prvky rozložení . . . . .	12
3.4	Základní ovládací prvky . . . . .	14
3.5	Další prvky grafického prostředí . . . . .	17
3.6	Šablony a styly . . . . .	18
3.7	Data Binding . . . . .	19
3.8	Animace . . . . .	20
3.9	Webové služby . . . . .	22
<b>4</b>	<b>Screencasty</b>	<b>24</b>
<b>5</b>	<b>Ukázkové aplikace</b>	<b>26</b>
5.1	Kvíz . . . . .	26
5.2	Video přehrávač . . . . .	31
5.3	Předpověď počasí . . . . .	35
<b>6</b>	<b>Závěr</b>	<b>38</b>
<b>7</b>	<b>Reference</b>	<b>39</b>
	<b>Přílohy</b>	<b>40</b>
<b>A</b>	<b>Přílohy dokumentu</b>	<b>40</b>
A.1	Datový XML soubor počasí . . . . .	40
<b>B</b>	<b>Přílohy</b>	<b>42</b>

## Seznam obrázků

1	Logo platformy Microsoft Silverlight . . . . .	9
2	Ukázka vykreslené křivky pomocí syntaxe Path Markup Syntax . . . . .	11
3	Prvek Grid . . . . .	12
4	Canvas . . . . .	13
5	StackPanel . . . . .	13
6	Prvek ScrollViewer . . . . .	14
7	TextBox . . . . .	15
8	Button . . . . .	15
9	ComboBox . . . . .	15
10	ListBox . . . . .	16
11	CheckBox . . . . .	16
12	RadioButton . . . . .	16
13	Typy animací . . . . .	20
14	Animace1 . . . . .	21
15	Animace 2 . . . . .	21
16	Ukázka aplikace Screen Capture . . . . .	24
17	Ukázka části scénáře . . . . .	25
18	Aplikace Kvíz . . . . .	26
19	Aplikace Video přehrávač . . . . .	31
20	Aplikace Předpověď počasí . . . . .	35
21	Graf návštěvnosti kurzu za poslední půl rok . . . . .	38



## Seznam výpisů zdrojového kódu

1	Ukázka jazyku XAML . . . . .	11
2	Ukázka Path Markup Syntax . . . . .	11
3	Příklad šablony stylu prvku TextBox . . . . .	18
4	Ukázka nastavení stylu prvku TextBox . . . . .	18
5	Deklarace datové struktury a bindování . . . . .	19
6	Ukázka deklarace Bindování hodnot v XAML . . . . .	19
7	Definice Animace1 . . . . .	21
8	Definice Animace2 s použitím DoubleAnimationUsingKeyFrames . . . . .	22
9	Struktura XML volání metody pomocí WCF . . . . .	23
10	Struktura XML odpovědi metody pomocí WCF . . . . .	23
11	Struktura datového XML aplikace Kvíz . . . . .	27
12	Metoda ParseX . . . . .	28
13	Metoda Answer . . . . .	29
14	Deklarace parametrů aplikace Quiz v HTML definici . . . . .	30
15	Struktura souboru s titulky . . . . .	32
16	Parsování načteného souboru s titulky . . . . .	33
17	Vložení záznamů titulků do prvku MediaElement . . . . .	33
18	Metoda vyvolaná zjištěním záznamu titulku . . . . .	34
19	Deklarace parametrů aplikace video přehrávače v HTML definici . . . . .	34
20	Struktura XML nalezených lokací . . . . .	36
21	Metoda ParseCitys . . . . .	37
22	Úryvek těla metody MoveDayPanel . . . . .	37
23	Struktura XML detailu nalezené lokace . . . . .	40

## 1 Úvod

Od samého počátku, kdy se internet dostal do podvědomí veřejnosti, se vývojáři snažili zavést webové platformy a frameworky. Tyto platformy měli částečně či úplně nahrazovat desktopové aplikace a zaručit tak multiplatformnost a snadnou šířitelnost těchto aplikací. Z počátku se jednalo o jednoduché aplikace, ale vývojem prohlížečů se začaly objevovat čím dál složitější frameworky a platformy. Jednou z prvních platform umožňující vytvoření složitějších aplikací běžících v prohlížeči byl Java Applet, ten byl však brzo ohrožován a následně nahrazen platformou Macromedia Flash, která je dnes nejrozšířenější webovou platformou vůbec, a to díky snadné tvorbě a jazyku ActionScript, který vychází z JavaScriptu. V roce 2007 přišel Microsoft se svou webovou platformou Silverlight. Z počátku byl Silverlight stejně jako Flash postavený na JavaScriptu, ale časem přidali vývojáři Microsoftu své platformě možnost programování v jazyce C# nebo VB, a tím se stal Silverlight použitelnějším a rozšířenějším.

Platforma Microsoft Silverlight, stejně jako technologie Adobe Flash spadá do takzvaných RIA aplikací. RIA aplikace jsou takové aplikace, které běží ve webovém prohlížeči za pomoci pluginu a nabízí některé vlastnosti desktopových aplikací. Mezi RIA aplikace se řadí nejen již zmíněný Silverlight a Flash/Flex, ale také Java Applet a HTML ve spojení s JavaScriptem. RIA aplikace umožňují rychlý běh sama sebe, aniž by ovlivňovala své okolí, to znamená, že se aplikace stáhne jen jednou, a každá změna v aplikaci nepotřebuje obnovení stránky. Toto také umožňuje rychlejší interakci mezi uživatelem a aplikací, kde uživatel nemusí například čekat na potvrzení změn. Další výhodou RIA aplikací je offline provoz, který umožní uživateli pracovat s částmi aplikace, aniž by byla připojena k internetu či síti. Další výhodou je že RIA aplikace běží na straně klienta, a tak zmenšuje vytížení serveru. Problém však může nastat u mobilních zařízeních, kde klientské zařízení nenabízí prostředky jako desktop [8].

Platforma Silverlight spolu s Java Applety disponuje oproti platformě Flash větší funkcí, a to díky jazykům C# u Silverlightu a Javy u Java Appletu. Ale Flash naproti tomu disponuje mnohem širšími možnostmi grafického návrhu jak prostředí aplikace,

tak animací [3]. V dnešní době jsou převážně Silverlight a Flash vzájemnými konkurenty v rámci RIA aplikací.

Na internetu se nalézají mnoho tutoriálů a návodů k platformě Silverlight, jsou však v anglickém jazyce nebo jsou zastaralé. Existují sice také návody v českém nebo slovenském jazyce, ale tyto návody nejsou většinou komplexní, to znamená, že se zaměřují na problematiku některých vlastností platformy Silverlight, a ne na platformu jako takovou.

Tato práce pojednává o platformě Silverlight, a jejím obsahem bylo popsat tuto platformu a rozšířit stávající kurz o další kapitoly a interaktivní části, jako jsou testovací kvíz a video-návody, tak aby kurz umožnil čtenáři pochopení této platformy a naučil se základům a schopnostem Silverlightu.

## 2 Výukový modul

### 2.1 Cíle

Cílem práce bylo rozšířit výukový kurz vztahující se k nejpoužívanější částem platformy Silverlight, dále pak vytvořit sadu screencastů, zobrazujících obsah výukových kapitol a jejich začlenění do kapitol kurzu. Následně vznikla aplikace VideoPlayer (Video přehrávač), umožňující přehrávání těchto screencastů. Dále vznikla aplikace Quiz (Kvíz), jako test znalostí z jednotlivých kapitol kurzu. A jako třetí ukázková aplikace vznikla aplikace s názvem WeatherApp (Předpověď počasí), která reprezentuje ukázkou základních vlastností a schopností platformy Silverlight. Zkombinováním všech těchto částí s již existujícími kapitolami kurzu, vznikl celek, který se nalézá na webu <http://silverlight.cs.vsb.cz>.

#### Základní cíle práce:

- Rozšíření a úprava již existujícího kurzu
- Vytvoření série screencastů vztahující se k jednotlivým kapitolám
- Vytvoření aplikace Kvíz, sloužící jako ověření znalostí z jednotlivých kapitol
- Vytvoření aplikace VideoPlayer k přehrávání screencastů
- Vytvoření ukázkové aplikace předpovědi počasí, jakožto ukázkou schopností platformy Silverlight

### 2.2 Tvorba kapitol

Celý kurz byl rozdělen do dvou sekcí. V první sekci se nalézají výukové kapitoly, v druhé sekci pak kapitoly týkající se tří vytvořených aplikací. Při tvorbě nových kapitol webového kurzu se vycházelo ze stylu již existujících kapitol kurzu, pokračovalo se tedy v projektu ASP.NET dodaným vedoucím práce a zveřejněným na webu <http://silverlight.cs.vsb.cz>.

Samotný postup tvorby jednotlivých kapitol webového kurzu první sekce byl rozdělen do tří fází. V první fázi se rozhodlo jaké části platformy Silverlight bude kapitola

obsahovat, v druhé fázi se vytvářely mini-aplikace vysvětlující látku kapitoly. V třetí fázi se tvořil obsah kapitoly, tedy teoretická část a popis vytvořených programů, vkládání screenshotů a screencastů, případně kvízu.

Tato práce zahrnovala vytvoření kapitol 8 a 9, tedy kapitol týkajících se Webových služeb, Režimu „Out Of Browser“ a dalších schopností platformy Silverlight. Obsah těchto kapitol byl vybrán z chybějících základních vlastností platformy. Dále práce zahrnovala vytvoření kapitol týkajících se vytvořených aplikací a popisují jejich základní vlastnosti a důležité části kódu.

## 2.3 Tvorba screencastů

Screencast je videozáznam počítačové obrazovky obsahující audio doprovod popisující děj záznamu. Screencast umožňuje snazší pochopení návodu, kde je jasně ukázáno, kde se kód nalézá a k čemu slouží. Umožňuje tedy lepší orientaci v kódu a zvyšuje efektivitu pochopení probíraného kódu či kapitoly.

Existuje několik nástrojů umožňujících tvorbu screencastů, mezi ně patří například Expression Encoder 4 Screen Capture z balíčku Microsoft Expression, Adobe Captivate, Jing, CamStudio nebo Fraps. Všechny tyto aplikace umožňují záznam dění na obrazovce, ale aplikace Expression Encoder 4 Screen Capture se jevila jako nejlepší volba, a to nejen z důvodu velikosti výstupního souboru, ale také z důvodu integrace s balíčkem Expression, který nabízí mimo jiné nástroje pro úpravu videa Expression Encoder.

Jako výchozí kontejner byl zvolen soubor WMV, který umožňuje přehrávat prvek MediaElement využívaný v aplikaci Video přehrávač. S ohledem na velikost výstupního souboru a kvalitu záznamu bylo zvoleno rozlišení videozáznamu 1280x720 pixelů při 15fps, konstantní bitrate 1045Kbps a kodek VC-1 Advanced. Jako audio kodek byl zvolen WMA Professional s konstantním bitrate 192Kbps.

Jako nástroj pro audio záznam byla zvolena aplikace Audacity, která nabízela vše co bylo zapotřebí. Hrubý audio záznam byl pořizován ve formátu WAV.

## **2.4 Tvorba ukázkových aplikací**

V tvorbě aplikací se využily kapitoly kurzu ale také další možnosti Silverlightu. Zde bylo záměrem vytvořit aplikace, které se využijí v rámci výukového kurzu a pomohou s pochopením daného obsahu jednotlivých kapitol. Postup tvorby jednotlivých aplikací se nalézá v kapitole 5. Ke každé aplikaci jsou k dispozici zdrojové kódy s popisem a vysvětlením.

### 3 Platforma Microsoft Silverlight

Následující kapitoly představují část výstupu výukového materiálu, sloužící pro lepší představu obsahu práce a vlastností platformy Silverlight.

**Silverlight** je aplikační platforma vytvořená společností Microsoft, určena k vývoji business a multimediálních aplikací RIA. Aplikace této technologie fungují pomocí pluginu v prohlížeči nebo jako samostatná aplikace v režimu „Out Of Browser“ [5].

K vývoji aplikací Silverlight se využívá jazyk XAML pro návrh grafického prostředí a programovací jazyky C#, VB.NET nebo IronPython pro logickou část aplikace. Jako vývojové prostředí se primárně využívá Visual Studio nebo Expression Blend, ale také lze využít alternativu Eclipse4SL pro vývojové prostředí Eclipse, nebo vývojové prostředí MonoDevelop.



Obrázek 1: Logo platformy Microsoft Silverlight

Silverlight je také využíván spolu s XNA jako výchozí programová platforma pro operační systém Windows Phone 7. Zde je však Silverlight opatřen nadstavbou pro WP7 jako jsou nová rozložení, ovládací prvky a přístup k vlastnostem systému.

K platformě Silverlight existuje alternativa ve formě funkčně omezené nadstavbě platformy Mono s názvem Moonlight vyvíjená společností Novell, která je primárně

určena na operační systémy linux a její aktuální verze je funkčně srovnatelná s verzí Silverlightu 3 a částečně 4.

### **3.1 Základní vlastnosti platformy Silverlight**

- Možnost vyvíjet v jazycích C# nebo Visual Basic
- Vývojová prostředí Microsoft Visual Studio nebo Expression Blend
- Využívá jazyk XAML
- Podpora Bindingu, viz definice
- Možnost využití IsolatedStorage
- Aplikace běží v Sandboxu
- Podpora Linq to Objects
- Podpora ADO.NET frameworku
- Podpora WCF služeb
- Od verze 4.0 podpora tzv. drag'n'drop, tedy přetahování souborů
- Od verze 5.0 umožňuje využití XNA frameworku



## 3.2 Jazyk XAML

**Extensible Application Markup Language** je značkovací jazyk využívaný k popisu grafického rozhraní v technologiích společnosti Microsoft jako je Silverlight, WPF nebo WF. Jazyk XAML je založen na jazyku XML a je dostupný od .NET Frameworku verze 3.0 [5].

V následující ukázce je vidět definice prvku StackPanel, obsahující Button (tlačítko) a TextBox (textové pole). V hlavičce UserControl se nalézají veškeré potřebné definice jazyka XAML a jmenných prostorů.

```

1 <UserControl x:Class="SilverlightApplication3.MainPage"
2   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6   mc:Ignorable="d"
7   d:DesignHeight="300" d:DesignWidth="400">
8   <Grid x:Name="LayoutRoot" Background="White">
9       <StackPanel x:Name="MyStackPanel" Margin="50">
10          <Button x:Name="MyButton" Content="Button" Click="MyButton_Click" />
11          <TextBox x:Name="MyTextBox" Text="TextBox" />
12      </StackPanel>
13  </Grid>
14 </UserControl>

```

Výpis 1: Ukázka jazyku XAML

Jazyk XAML poskytuje také speciální syntaxi pro kreslení křivek nazvanou Path Markup Syntax. Tato syntaxe umožňuje rychle a snadno vykreslit objekt typu Path, například Bézierovu křivku.



Obrázek 2: Ukázka vykreslené křivky pomocí syntaxe Path Markup Syntax

```

1 <Path Stroke="Black" Fill="Gray" Data="M 10,100 C 10,300 300,-200 300,100" />

```

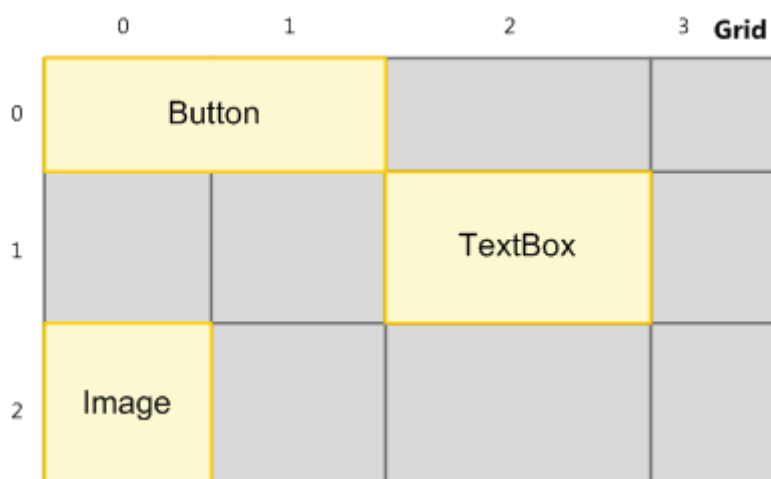
Výpis 2: Ukázka Path Markup Syntax

### 3.3 Základní prvky rozložení

Platforma Silverlight nabízí čtyři základní prvky rozložení: Grid, Canvas, Stackpanel a Border.

#### 3.3.1 Grid

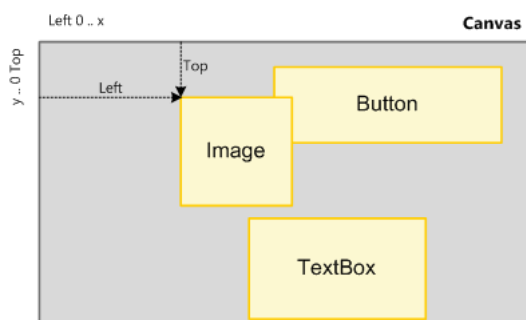
**Grid** je základním a nejčastěji používaným rozložením. Jedná se o mřížku, která je definována vlastnostmi `Grid.ColumnDefinition` a `Grid.RowDefinition`, tedy sloupce a řádky. Rozměry sloupců a řádků je možné zadávat relativně nebo absolutně.



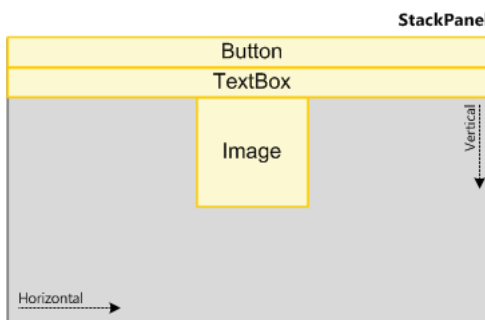
Obrázek 3: Prvek Grid

#### 3.3.2 Border

**Border** je komponenta grafického prostředí, která má jediný účel, a to graficky ohraničit vložený prvek.



Obrázek 4: Canvas



Obrázek 5: StackPanel

### 3.3.3 Canvas

**Canvas** (Obr. 4) je nejvolnějším rozložením, chová se jako kreslící plátno, na němž se prvky rozmístí ují pomocí vlastností `Canvas.Left` a `Canvas.Top`. Prvek `Canvas` nám primárně umožňuje kreslení tvarů jako je elipsa, obdelník, nebo různé křivky, ale umožňuje také vkládání ostatních prvků grafického prostředí.

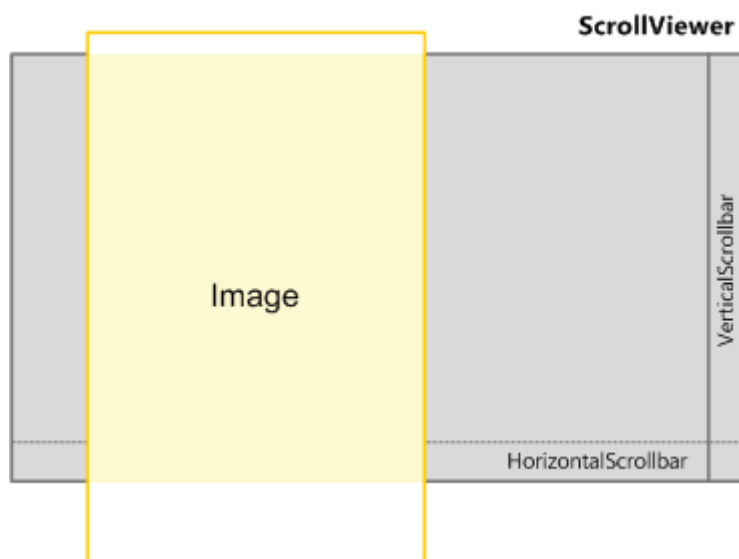
### 3.3.4 StackPanel

**StackPanel** (Obr. 5) je druhý nejpoužívanější Layout po prvku `Grid`. `StackPanel` vkládá prvky buďto vertikálně (vlastnost `Orientation` je nastavena na hodnotu `Vertical`, což je také výchozí nastavení), nebo horizontálně (vlastnost `Orientation` je nastavena na hodnotu `Horizontal`).

## 3.4 Základní ovládací prvky

### 3.4.1 ScrollViewer

**ScrollViewer** je velmi užitečná a často používaná komponenta, sloužící k zobrazování objektů, větších než ScrollViewer sám. Může obsahovat pouze jeden jediný další prvek. ScrollViewer se často využívá ve spojení s komponentou Image nebo StackPanel.



Obrázek 6: Prvek ScrollViewer

### 3.4.2 TextBlock

**TextBlock** reprezentuje obyčejné výstupní textové pole. Existuje alternativa ve formě prvku **Label**, který se využívá ke stejným účelům, ale na rozdíl od prvku **TextBlock** nemá vlastnost **Text** ale **Content**.

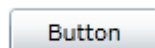
### 3.4.3 TextBox

**TextBox** (Obr. 7) je velice často používanou komponentou, jedná se o vstupní textové pole. Existuje také prvek kombinující vlastnosti **TextBoxu** a **ListBoxu** nebo třeba **DataGridu**.

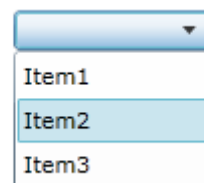
Tento prvek se nazývá `AutoCompleteBox` a funguje jako našeptávač, kde se postupným psaním objevují návrhy.



Obrázek 7: TextBox



Obrázek 8: Button



Obrázek 9: ComboBox

#### 3.4.4 Button

**Button** (Obr. 8) „česky tlačítko“, je základní ovládací prvek grafického prostředí platformy Silverlight. Jedná se o obyčejné tlačítko, kterému lze snadno změnit vzhled jako každému prvku pomocí šablon. Prvek `Button` může místo textu v atributu `Content` obsahovat jiný prvek nebo rozložení, to umožňuje tlačítku přizpůsobovat vzhled a měnit možnosti jeho využití.

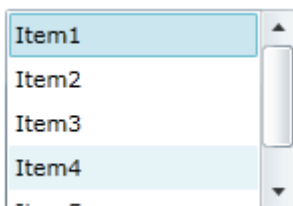
Existuje také speciální druh tlačítka, které poskytuje navíc stav `Checked`, takovéto tlačítko se nazývá `ToggleButton`. `ToggleButton` nabývá dvou respektive tří hodnot ve vlastnosti `IsChecked`, a to `true` (pravda), `false` (nepravda). Může také nabývat stavu `indeterminate` tedy neurčitého stavu.

#### 3.4.5 ComboBox

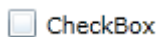
**ComboBox** (Obr. 9) slouží k zobrazení seznamu položek, ze kterých uživatel vybírá jednu položku. Pokud je seznam možností větší než počet zobrazovaných možností objeví se posuvník, který umožňuje posouvání mezi ostatními položkami seznamu.

#### 3.4.6 ListBox

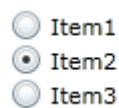
**ListBox** (Obr. 10) slouží k zobrazení seznamu položek, ze kterých může uživatel vybírat jednu nebo více položek.



Obrázek 10: ListBox



Obrázek 11: CheckBox



Obrázek 12: RadioButton

### 3.4.7 CheckBox

**CheckBox** (Obr. 11) reprezentuje zaškrtnuté pole. **CheckBox** nabývá dvou respektive tří hodnot, a to `true` (vybráno), `false` (nevybráno). Může také nabývat stavu `indeterminate` tedy neurčitého stavu.

### 3.4.8 RadioButton

**RadioButton** (Obr. 12) sám o sobě nemá význam, ale ve větším počtu reprezentuje přepínač mezi více položkami.

### 3.4.9 MediaElement

**MediaElement** je prvek umožňující přehrávání audia a videa v platformě Silverlight.

#### Podpora formátů prvku **MediaElement** Kontejner - Kodek

- MP3 - Audio: MP3
- ASF - Audio: WMA; Video: WMV, VC-1
- MP4 - Audio: AAC-LC, HE-AAC v1 (AAC+), HE-AAC v2 (eAAC+), MPEG-4 Part 2\*; Video: H.264 | \* podoprováno jen v WP7
- 3GP\*, 3G2\* - Audio: AAC-LC, HE-AAC v1 (AAC+), HE-AAC v2 (eAAC+), AMR-NR; Video: H.264, H.263 | \* podoprováno jen v WP7

### 3.5 Další prvky grafického prostředí

Mezi další prvky patří i ProgressBar (ukazatel stavu), Slider (posuvný potenciometr), DataGrid (Mřížka zobrazující data, například z databáze), různé dialogy, jako například FileDialog nebo MessageBox, mnoho dalších komponent a prvků.

Další prvky grafického prostředí nalezneme v Silverlight Toolkitu, který se nalézá na stránkách <http://silverlight.codeplex.com/>. Tento toolkit nabízí například kolekci různých grafů, kontextová menu a mnoho dalších prvků.

### 3.6 Šablony a styly

Díky jazyka XAML lze na každý prvek snadno aplikovat vlastní styl. Styly představují způsob, jak vytvořit seznam vlastností pro typ prvku.

Styly jsou definovány v elementu `Resources` prvku. Každý element ve stromu prvků může mít svou `Resources` sekci.

Dále můžeme styly umístit na úrovni `Application` (tedy globálně), v `UserControl` nebo v `ResourceDictionary`, která představuje elegantní způsob správy stylů oddělených od ostatní deklarací stylů. `ResourceDictionary` pak ale musí být připojena na potřebné místo pomocí `MergeDictionaries` [4].

Styly jsou ukládány do `Dictionary`, a proto musí mít každý styl nadefinován atribut `x:Key`, tedy klíč stylu, nebo lze použít implicitní přiřazení na základě typu prvku, jehož se styl týká, k čemuž slouží vlastnosti `TargetType`.

```

1 <UserControl.Resources>
2   <Style x:Key="ErrorStyle" TargetType="TextBox">
3     <Setter Property="FontSize" Value="16" />
4     <Setter Property="FontWeight" Value="Bold" />
5     <Setter Property="Foreground" Value="Red" />
6   </Style>
7 </UserControl.Resources>

```

Výpis 3: Příklad šablony stylu prvku `TextBox`

```

1 <StackPanel>
2   <TextBox Text="TextBox" />
3   <TextBox Text="TextBox ErrorStyle" Style="{StaticResource ErrorStyle}" />
4 </StackPanel>

```

Výpis 4: Ukázka nastavení stylu prvku `TextBox`

Veškeré styly a šablony se nalézají na stránkách MSDN [7] a nebo je lze získat rozložením prvku v prostředí Expression Blend.



### 3.7 Data Binding

**Data Binding** je způsob datového propojení uživatelského rozhraní s logikou aplikace. Data Binding tedy snadno za pomoci objektů umožňuje přenášet data do grafických prvků. Toto lze využít například u vyplňování formulářů nebo lze přenášet výsledky z databáze do prvků jako je DataGrid nebo ListBox. Mechanismus, který řídí komunikaci mezi zdrojem dat a uživatelským rozhraním, může být například rozhraní `INotifyPropertyChanged` nebo například událost či `DependencyProperty`.

Za zmínku také stojí `DataTemplate`, tedy šablona zobrazování dat získaných pomocí Data Bindingu. Příkladem použití `DataTemplate`, tedy datových šablon, je řádek seznamu obsahující několik údajů v předdefinovaném formátu.

V následujícím ukázce se nalézá definice třídy `Data`, reprezentující data formuláře. Instance této třídy se předá pomocí Data Bindingu a vlastnosti `DataContext` uživatelskému rozhraní.

```

1 public class Data{
2     public string Text {set; get; }
3     public string ButtonText {set; get; }
4 } ...
5 Data data = new Data(){
6     Text = "Text TextBoxu", ButtonText = "Text prvku Button"
7 };
8 this.DataContext = data;
```

Výpis 5: Deklarace datové struktury a bindování

V následujícím XAML kódu je použito Bindování hodnot do grafických prvků. Vlastnost `DataContext` přiřadí všechny existující vlastnosti označené `Binding` k vlastnostem přijatého objektu, v tomto případě instanci třídy `Data`, a vloží do nich patřičná data.

```

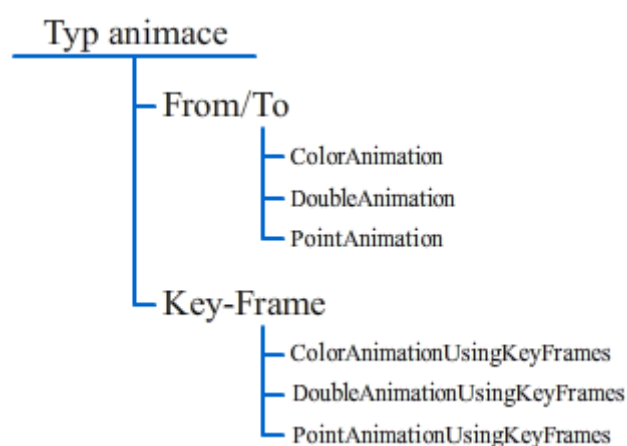
1 <Grid x:Name="LayoutRoot" Background="White">
2     <StackPanel Margin="50">
3         <Button x:Name="MyButton" Content="{Binding ButtonText}"/>
4         <TextBox x:Name="MyTextBox" Text="{Binding Text}" />
5     </StackPanel>
6 </Grid>
```

Výpis 6: Ukázka deklarace Bindování hodnot v XAML

### 3.8 Animace

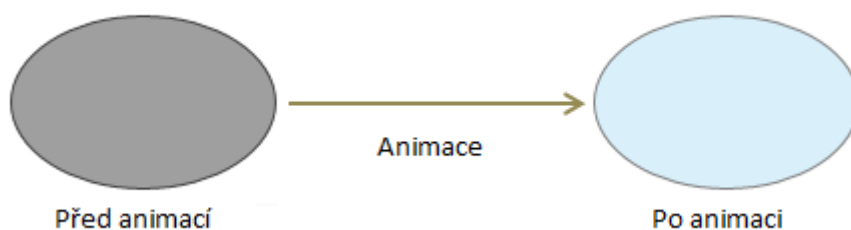
Platforma Silverlight umožňuje animovat prvky a vlastnosti ovládacích prvků a tvarů, k tomu využívá třídu `Storyboard`. `Storyboard` zaobaluje animace do pojmenované struktury. Animace `Storyboardu` lze spustit, pozastavit a nebo zastavit. Animace se definují buďto v `Resources` prvku nebo aplikace.

Existují dva typy animací, a to `From/To` a `Key-Frame`. `From/To` jsou animace vlastností prvků nabývajících hodnot od/do a lze zde využít i předdefinované animace zvané `Animation Easing`. `Key-Frame` animace jsou tvořeny `Framy`, tedy jednotlivými snímky, mezi nimiž se provádí jednotlivé animace.



Obrázek 13: Typy animací

Každá animace musí obsahovat informaci o prvku, kterého se animace týká, k tomu slouží vlastnost `TargetName`, a informaci o vlastnosti, kterou bude animace měnit, k tomu slouží vlastnost `TargetProperty`. V obou typech animací můžeme měnit tři druhy vlastností, a to barvu, vlastnosti typu `Double`, jako například `Width`, a nebo pozici bodu.



Obrázek 14: Animace1

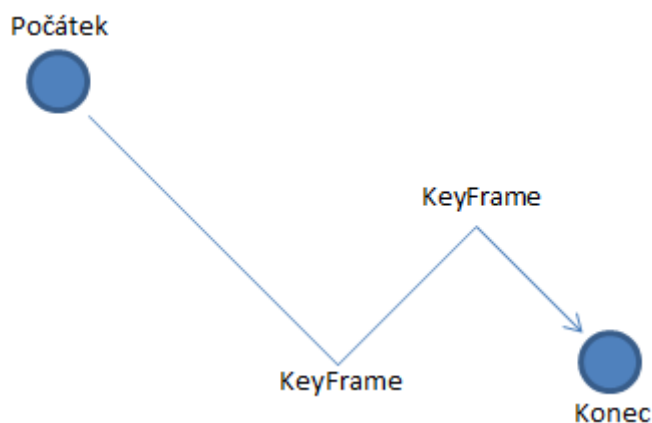
```

1 <Storyboard Name="Animace">
2   <ColorAnimation Storyboard.TargetName="Objekt" Storyboard.TargetProperty="(Shape.Fill).(
3     SolidColorBrush.Color)" Duration="0:0:0.2" To="#551ba1e2"/>
4   <DoubleAnimation Storyboard.TargetName="Objekt" Storyboard.TargetProperty="UIElement.
5     Opacity" To="0.5" Duration="0:0:0.2" />
6 </Storyboard>

```

Výpis 7: Definice Animace1

V XAML kódu se nalézá definice Storyboardu s dvěma animacemi typu From/To, které se provedou najednou. První animace je typu `ColorAnimation`, ta aplikuje změnu barvy vyplnění prvku Objekt v době trvání 20ms. Druhá animace je typu `DoubleAnimation`, která animuje změnu průhlednosti prvku Objekt v době trvání 20ms.



Obrázek 15: Animace 2

```

1  <Storyboard x:Name="MyStoryboard">
2      <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(Canvas.Left)"
3          Storyboard.TargetName="Objekt">
4          <LinearDoubleKeyFrame KeyTime="0" Value="0" />
5          <LinearDoubleKeyFrame KeyTime="0:0:0.6" Value="300" />
6          <LinearDoubleKeyFrame KeyTime="0:0:1" Value="400" />
7          <LinearDoubleKeyFrame KeyTime="0:0:1.3" Value="500" />
8      </DoubleAnimationUsingKeyFrames>
9      <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(Canvas.Top)"
10         Storyboard.TargetName="Objekt">
11         <LinearDoubleKeyFrame KeyTime="0" Value="0" />
12         <LinearDoubleKeyFrame KeyTime="0:0:0.6" Value="450" />
13         <LinearDoubleKeyFrame KeyTime="0:0:1" Value="300" />
14         <LinearDoubleKeyFrame KeyTime="0:0:1.3" Value="450" />
15     </DoubleAnimationUsingKeyFrames>
16 </Storyboard>

```

Výpis 8: Definice Animace2 s použitím DoubleAnimationUsingKeyFrames

V XAML kódu se nalézá definice Storyboardu s animací typu `DoubleAnimationUsingKeyFrames`. Animuje se ostrý odraz elipsy v určených souřadnicích.

### 3.9 Webové služby

Pojem webová služba se dnes často používá ve spojení s koncepcí SOA (Service Oriented Architecture), architektury orientovaných služeb. Pomocí SOA architektury můžeme tvořit distribuované systémy, poskytující funkčnost aplikace jako službu koncovému uživateli nebo dalším službám. Silverlight může využívat několika služeb, mezi ně patří webová služba WebClient a služba WCF.

WebClient slouží ke stahování souborů a je velmi hojně využíván v aplikacích typu RSS čteček, předpovědí počasí. WebClient se také využívá v aplikaci WeatherApp, která je součástí této práce.

WCF je služba sloužící ke komunikaci klientské aplikace s aplikací umístěnou někde na serveru. Služba WCF sjednocuje všechny předchozí služby technologie .NET frameworku a velice často se využívá ke vzdálené komunikaci klientské aplikace s databází. Komunikace probíhá pomocí XML dat, které se odesílají z klienta na server nebo naopak.

WCF služba může komunikovat buďto synchronně nebo asynchronně. Samotný postup komunikace mezi klientem a serverem za pomoci služby WCF probíhá přibližně takto:

- Zavolá se metoda služby.
- Služba na straně klienta převezme volání metody a přenese data získaná v parametrech metody do přenositelné formy, to znamená celou metodu s parametry přepíše do struktury XML.
- Služba odešle XML na severovou aplikaci.
- Serverová aplikace převezme XML a to následně deserializuje. Deserializovanou metodu zavolá a provede.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4   xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
5   <soap12:Body>
6     <HelloWorld xmlns="http://tempuri.org/" />
7   </soap12:Body>
8 </soap12:Envelope>

```

Výpis 9: Struktura XML volání metody pomocí WCF

```

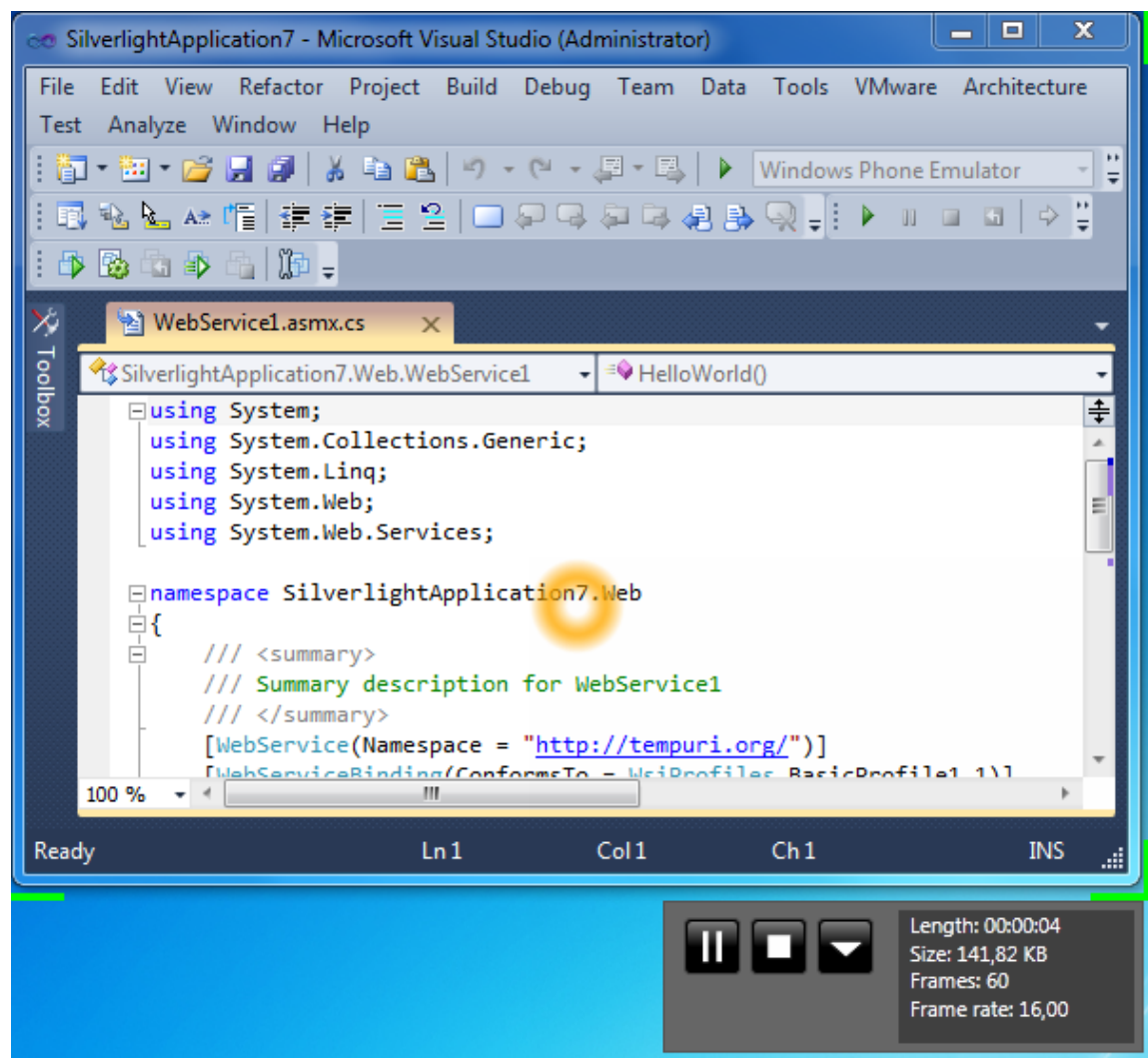
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
4   xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
5   <soap12:Body>
6     <HelloWorldResponse xmlns="http://tempuri.org/">
7       <HelloWorldResult>string</HelloWorldResult>
8     </HelloWorldResponse>
9   </soap12:Body>
10 </soap12:Envelope>

```

Výpis 10: Struktura XML odpovědi metody pomocí WCF

## 4 Screencasty

Postup tvorby screenacastů byl rozdělen do čtyř fází. V první fázi byl prováděn záznam částí kapitol, zde byla zvolena aplikace Microsoft Expression Encoder 4 Screen Capture.



Obrázek 16: Ukázka aplikace Screen Capture

V druhé fázi byl připravován scénář akcí, týkající se událostí dění v záznamu. Jednotlivé scénáře popisují děj videa, a byly vytvářeny po vytvoření screencastů.

Nyní přejdeme k logické části aplikace.

Vytvořila se událost `canvas_Drop`, která reprezentuje reakci na přetažení souborů.

Vytvoříme pole třídy `FileInfo`, nazveme ho `droppedFiles`.

Pole bude obsahovat informace o přetažených souborech do komponenty `Canvas`, získané událost `Drop`, které vrací v argumentu `e`.

Získané pole souborů budeme procházet příkazem `foreach`.

Napřed si však připravíme dvě metody.

První nazveme `kontrola` a bude vracet `boolean` a v argumentu bude přijímat koncovku souboru.

Metoda vrátí `true` pokud se bude jednat o obrázek typu `jpg`, `jpeg` nebo `png`.

### Obrázek 17: Ukázka části scénáře

Následně v třetí fázi byl prováděn audio záznam aplikací `Audacity` podle videozáznamu a scénáře. Audio záznam se prováděl za pomoci mikrofону u herních sluchátek. Kvalita vzniklého audio záznamu odpovídá průměrné kvalitě. Audio záznam probíhal během přehrávání screencastu, aby se docílilo synchronizace audia a videa.

V poslední, čtvrté fázi, probíhala úprava a střih audio a videozáznamu a jejich následné sjednocení v aplikaci `Expression Encoder`.

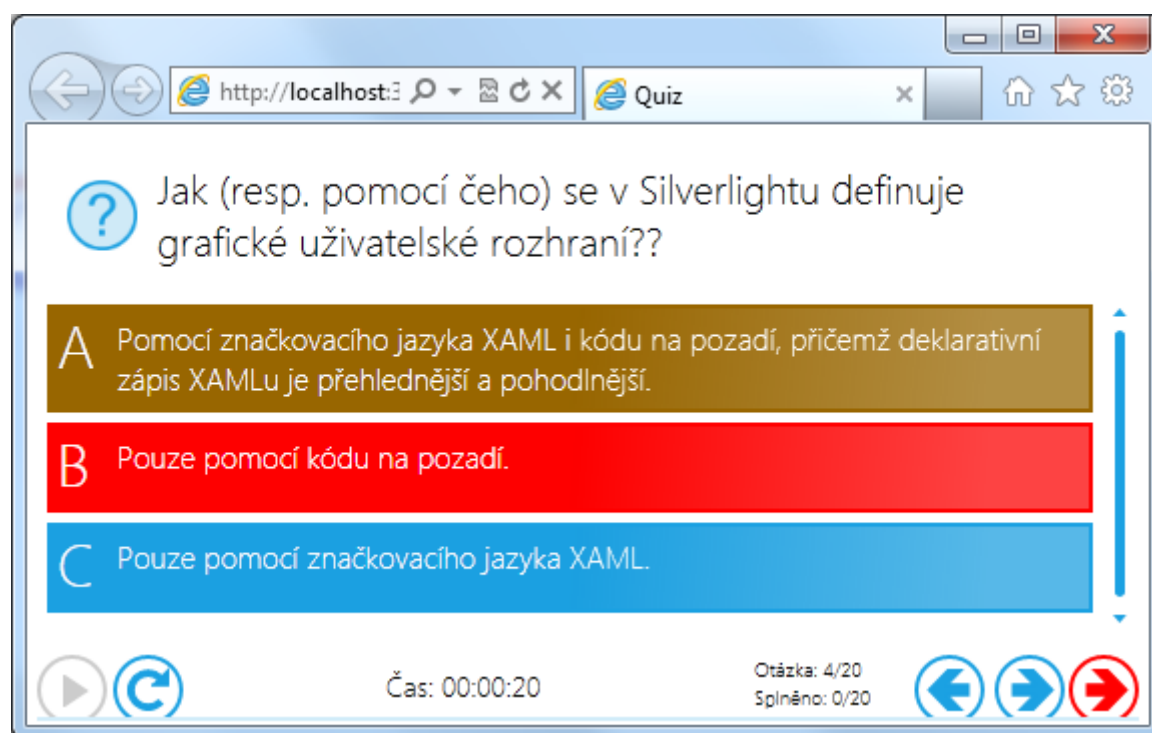
Takto vytvořené screencasty se pak zveřejnily v interaktivním kurzu za pomoci aplikace `VieoPlayer` (Video přehrávač).

## 5 Ukázkové aplikace

V rámci této bakalářské práce vznikly tři aplikace ukazující základní možnosti platformy Microsoft Silverlight.

### 5.1 Kvíz

Aplikace Kvíz (Quiz), je doprovodnou aplikací k výukovému modulu, která umožňuje přezkoušení z jednotlivých lekcí nebo celého kurzu. Tato aplikace načítá externí XML soubor, který poskytuje zdroj dat k přezkušování z jednotlivých lekcí kurzu. Zdrojový XML soubor obsahuje uvítací text, otázky a odpovědi k otázkám, a aplikace jej získá skrze parametr v HTML definici aplikace.



Obrázek 18: Aplikace Kvíz



**Požadavky:**

- Metro styl
- Snadné ovládání
- Načítání externího datového XML souboru
- Snadná úprava nebo přidání otázek kurzu
- Předávání parametrů aplikace skrze HTML definici
- Možnost více správných odpovědí

Tvorba aplikace Kvíz byla rozdělena do několika fází. V první fázi se tvořil hrubý návrh grafického prostředí, tak aby aplikace byla přehledná a snadno ovladatelná. V druhé fázi se vytvářela struktura datového XML souboru obsahujícího otázky a jejich odpovědi. V následující fázi se vytvářela knihovna obsahující objektovou reprezentaci XML souboru a metody zpracovávající data, tedy metody pro stahování a parsování datového XML souboru. V poslední fázi se realizoval návrh grafického prostředí a logika aplikace, využívající vytvořenou knihovnu.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <quiz>
3    <intro>
4      <text1>Uvitani 1</text1>
5      <text2>Uvitani 2</text2>
6    </intro>
7    <questions>
8      <question lesson="2">
9        <text>Text otazky z~lekce 2</text>
10       <answers>
11         <answer isCorrect="False">Nespravna odpoved</answer>
12         <answer isCorrect="False">Nespravna odpoved</answer>
13         <answer isCorrect="True">Spravna odpoved</answer>
14       </answers>
15     </question>
16   </questions>
17 </quiz>

```

Výpis 11: Struktura datového XML aplikace Kvíz

- Element **quiz** je hlavní element datového XML souboru
- Element **intro** obsahuje uvítací texty

- Element **questions** obsahuje všechny otázky ze všech kapitol
- Element **question** reprezentuje jednu otázku
  - Parametr **lesson** reprezentuje číslo lekce, které se otázka týká
- Tag **text** v elementu **question** reprezentuje zadání otázky
- Element **answers** obsahuje seznam odpovědí
- Tag **answer** reprezentuje jednu odpověď
  - Parametr **isCorrect** nastavuje pravdivost otázky

Mezi zajímavé části kódu aplikace jistě patří metoda `ParseX\verb`. Tato metoda provádí parsování získaného datového XML souboru do datové struktury, a využívá dotazovací jazyk LINQ (LINQ to Objects), pomocí kterého probíhá samotné parsování. Získaná datová struktura se vloží do argumentu události `ParseComplete`, která je vyvolána po dokončení parsování.

```

1 public void ParseX(string dataa){
2     XDocument x = XDocument.Parse(dataa);
3     Intro intro = (from data in x.Descendants("intro") select new Intro{
4         Text1 = data.Element("text1").Value,
5         Text2 = data.Element("text2").Value
6     }).ToList() [0];
7     List<Item> itemList = (from data in x.Descendants("question") select new Item{
8         Lesson = data.Attribute("lesson").Value,
9         Question = data.Element("text").Value,
10        AnswersT = (from dat in data.Descendants("answer")
11            where dat.Attribute("isCorrect").Value.Equals("True") select new Answer{
12                answer = dat.Value,
13                isTrue = true
14            }).ToList() ,
15        AnswersF = (from dat in data.Descendants("answer")
16            where dat.Attribute("isCorrect").Value.Equals("False") select new Answer{
17                answer = dat.Value,
18                isTrue = false
19            }).ToList()
20        }).ToList() ;
21     items.AddItems(itemList);
22     QuizGeneratedArgumentClass args = new QuizGeneratedArgumentClass();
23     args.Msg = "Parse complete";
24     args.Items = items; args.Intro = intro ;
25     if (ParseComplete!=null) ParseComplete(this, args);
26 }

```

Výpis 12: Metoda ParseX

Další zajímavou metodou je metoda `Answer`. Ta kontroluje, zda vybraná odpověď `answer` otázky `item` je správná nebo ne. Nejprve se zkontroluje, zda vybraná odpověď již nebyla počítána, následně se procházejí všechny odpovědi otázky a pokud je odpověď správná a byly zodpovězeny všechny odpovědi je vyvolána událost `QuestionComplete`. Ve chvíli, kdy se dokončí kontrola správnosti odpovědi a není otázka plně zodpovězena, vyvolá se událost `AnswerIs`, nesoucí informaci, zda odpověď byla nebo nebyla správná.

```

1  public void Answer(object Source, object Source2, Item item, Answer answer){
2      bool selected = false;
3      if (item.TrueAnswersCount != 0){
4          QuizArgumentClass arg = new QuizArgumentClass();
5          if (!answer.IsSelected){
6              foreach (Answer a in item.Answers){
7                  if (a.IsTrue == answer.IsTrue && a.IsTrue){
8                      selected = true;
9                      arg.Answer = answer;
10                     arg.Source = Source;
11                     arg.AnswerIs = true;
12                     item.TrueAnswersCount--;
13                     if (item.TrueAnswersCount == 0){
14                         QuizCompleteArgumentClass arg2 =
15                             new QuizCompleteArgumentClass();
16                         arg2.Item = item;
17                         if (QuestionComplete != null)
18                             QuestionComplete(Source2, arg2);
19                     }
20                     break;
21                 }else{
22                     selected = true;
23                     arg.Answer = answer;
24                     arg.Source = Source;
25                     arg.AnswerIs = false;
26                 }
27             }
28             arg.Answer.IsSelected = selected;
29             if (arg.AnswerIs) item.Attempts++;
30             else item.Attempts++;
31             arg.Item = item;
32             if (AnswerIs != null)
33                 AnswerIs(Source, arg);
34         }else
35             MessageBox.Show("FATAL");
36     }
37 }

```

Výpis 13: Metoda Answer

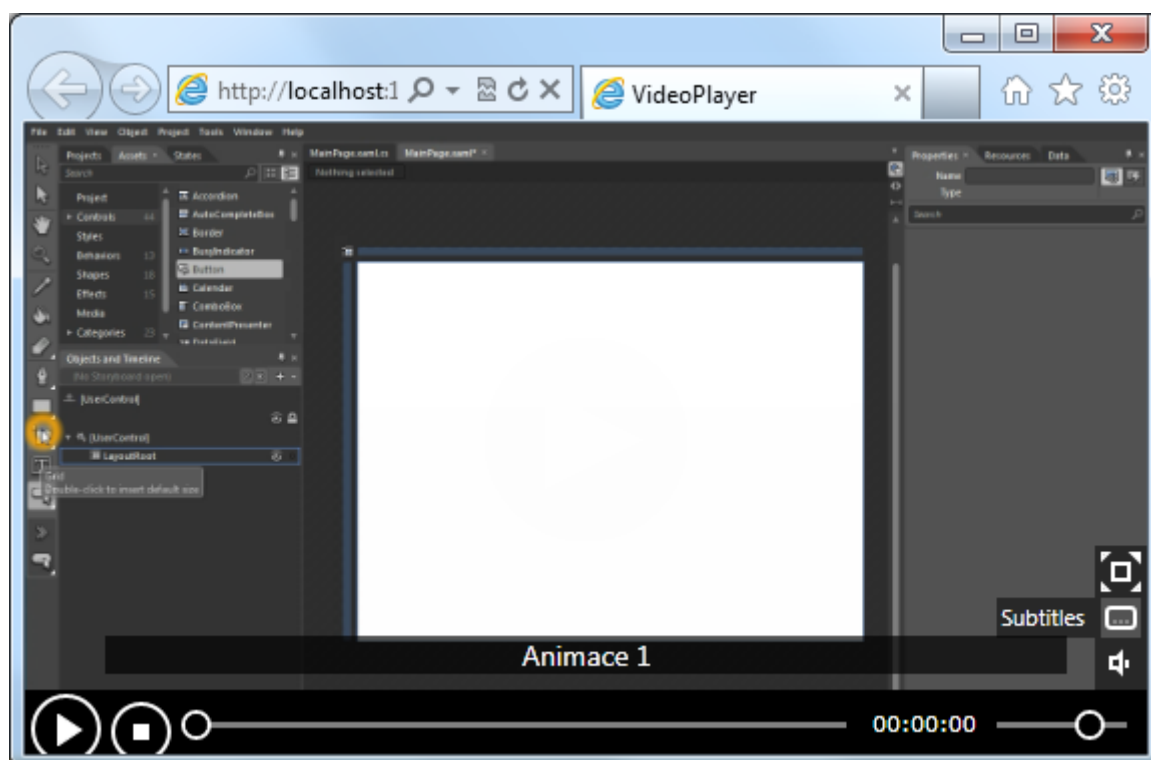
V následujícím kódu HTML deklarace aplikace se nalézá tag `param`, který nastavuje čísla zobrazovaných lekcí a adresa ke zdrojovému souboru s otázkami. Pokud bude atribut `lessons` tohoto tagu nastaven na hodnotu `-1`, bude aplikace zobrazovat všechny otázky ze všech lekcí.

```
1 <div id="silverlightControlHost">
2   <object data="data:application/x-silverlight-2," type="application/x-silverlight-2"
3     width="600" height="300">
4     <param name="source" value="ClientBin/Quiz.xap"/>
5     <param name="onError" value="onSilverlightError" />
6     <param name="background" value="white" />
7     <param name="minRuntimeVersion" value="4.0.50826.0" />
8     <param name="autoUpgrade" value="true" />
9
10    <!-- Definice parametru aplikace -->
11    <param name="InitParams" value="Lesson=-1,URL=http://localhost:3333/QuizData.xml" />
12
13    <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50826.0"
14      style="text-decoration:none">
15      
17    </a>
18  </object>
19  <iframe id="_sl_historyFrame" style="visibility:hidden;height:0px;width:0px;border:0px" />
20</div>
```

Výpis 14: Deklarace parametrů aplikace Quiz v HTML definici

## 5.2 Video přehrávač

Aplikace Video přehrávač (VideoPlayer) slouží, k přehrávání vytvořených screencastů. Aplikace umožní přehrávání video souborů podporovaných technologií Microsoft Silverlight, dále pak umožní načítání externího souboru s titulky. Veškerá data aplikace získává skrze adresy zadané v HTML definici.



Obrázek 19: Aplikace Video přehrávač

**Požadavky:**

- Metro styl
- Snadné ovládání
- Využití prvku `MediaElement`
- Načítání externího souboru s titulky
- Předávání parametrů aplikace skrze HTML definici

Návrh aplikace byl rozčleněn do tří fází. V první fázi se vytvářel hrubý grafický návrh aplikace, tak aby ovládání bylo přehledné. V druhé fázi se navrhovala a realizovala struktura souboru s titulky. V třetí fázi se realizovala samotná aplikace podle grafického návrhu a její logika.

```
1  0|10| Prilis  zlutoucky kun
2  20|25|upel
3  30|34|dabelske ody.
```

**Výpis 15: Struktura souboru s titulky**

Každý řádek reprezentuje jeden záznam. Každý záznam obsahuje 3 elementy oddělené znakem `'|'`.

- 1. element reprezentuje čas kdy se titulek zobrazí
- 2. element reprezentuje čas kdy se titulek skryje
- 3. element je samotný text titulku

Titulky se parsují pomocí jednoduché metody vyvolané událostí `OpenReadCompleted` instancí třídy `WebClient`. V této metodě probíhá přečtení a následné parsování souboru s titulky. Po přečtení souboru se vytvoří pole řádků záznamů rozdělených podle znaku nového řádku. Toto pole se následně prochází a každý záznam se dělí podle znaku `'|'` na tři elementy. Vytvoří se instance třídy `TitleX`, do které se vloží získaná data. Tato instance se vloží do listu `List` a po dokončení procházení záznamů se vyvolá událost `OnComplete`.

```

1  void wc_OpenReadCompleted(object sender, OpenReadCompletedEventArgs e){
2      if (e.Error == null){
3          StreamReader sr = new StreamReader(e.Result.*, Encoding.UTF8*);
4          string [] subData = sr.ReadToEnd().Split("\n");
5          List = new List<TitleX>();
6          foreach (string s~in subData){
7              string [] temp = s.Split(' ');
8              TitleX item = new TitleX();
9              item.InTime = Convert.ToDouble(temp[0]);
10             item.OutTime = Convert.ToDouble(temp[1]);
11             item.Message = temp[2];
12             item.Id = Guid.NewGuid().ToString();
13             List.Add(item);
14         }
15         if (OnComplete != null) OnComplete(List);
16     }else
17         if (ErrorEvent != null) ErrorEvent(this, null);
18 }

```

Výpis 16: Parsování načteného souboru s titulky

Media element dokáže zpracovat uživatelsky vložené titulky. Nejprve je však nutné převést datovou strukturu titulků do srozumitelné formy MediaElementu. K tomu slouží metoda `AddMarker`. Tato metoda tedy provede konverzi, jsou-li titulky k dispozici.

```

1  private void AddMarker(){
2      if (HasSubtitles) {
3          foreach (TitleX comment in SubTitles.List){
4              this.Video.Markers.Add(new TimelineMarker(){
5                  Time = TimeSpan.FromSeconds(comment.InTime),
6                  Type = comment.Id.ToString() + ":Start",
7                  Text = comment.Message });
8              this.Video.Markers.Add(new TimelineMarker(){
9                  Time = TimeSpan.FromSeconds(comment.OutTime),
10                 Type = comment.Id.ToString() + ":End",
11                 Text = "" });
12          }
13      }
14  }

```

Výpis 17: Vložení záznamů titulků do prvku MediaElement

Obsahuje-li MediaElement titulky a zároveň vyvolá událost `MarkerReached`, tedy událost která je generovaná zjištěním, že se v čase aktuálního stavu videa nalézají záznam titulků, je zavolána metoda `Video_MarkerReached`. Tato metoda zjistí, zda získaný

záznam je typu 'Start' nebo 'End'. Pokud je typu 'Start', pak se titulek zobrazí, pokud typu 'End', pak se titulek skryje.

```

1 private void Video_MarkerReached(object sender, TimelineMarkerRoutedEventArgs e)
2 {
3     string [] array = e.Marker.Type.Split(':');
4     if (array != null && array.Length == 2){
5         if (array[1] == "Start"){
6             isSubtitleAviable = true;
7             SubTitle.Text = e.Marker.Text;
8             SubTitle.Visibility = System.Windows.Visibility.Visible;
9         }
10        else if (array[1] == "End"){
11            SubTitle.Visibility = System.Windows.Visibility.Collapsed;
12            SubTitle.Text = string.Empty;
13            isSubtitleAviable = false;
14        }
15    }
16 }

```

Výpis 18: Metoda vyvolaná zjištěním záznamu titulku

Vložení tagu `param` do HTML definice zajistí nastavení aplikace. V tomto případě se předává parametrem `Video` adresa k video souboru, nepovinným parametrem `Title` uvítací text a nepovinným parametrem `Subtitle` adresa k souboru s titulky.

```

1 <div id="silverlightControlHost">
2     <object data="data:application/x-silverlight-2," type="application/x-silverlight-2"
3         width="640" height="390">
4         <param name="source" value="ClientBin/VideoPlayer.xap"/>
5         <param name="onError" value="onSilverlightError" />
6         <param name="background" value="white" />
7         <param name="minRuntimeVersion" value="4.0.50826.0" />
8         <param name="autoUpgrade" value="true" />
9
10        <!-- Definice parametru aplikace -->
11        <param name="InitParams" value="Video=Video.wmv,Title=Animace 1,Subtitle=Subs.txt" />
12
13        <a href="http://go.microsoft.com/fwlink/?LinkId=149156&v=4.0.50826.0" style="text-decoration
14            :none">
15            
17        </a></object><iframe id="Iframe1" style="visibility :hidden;height:0px;width:0px;border:0px"></
18        iframe>
19    </div>

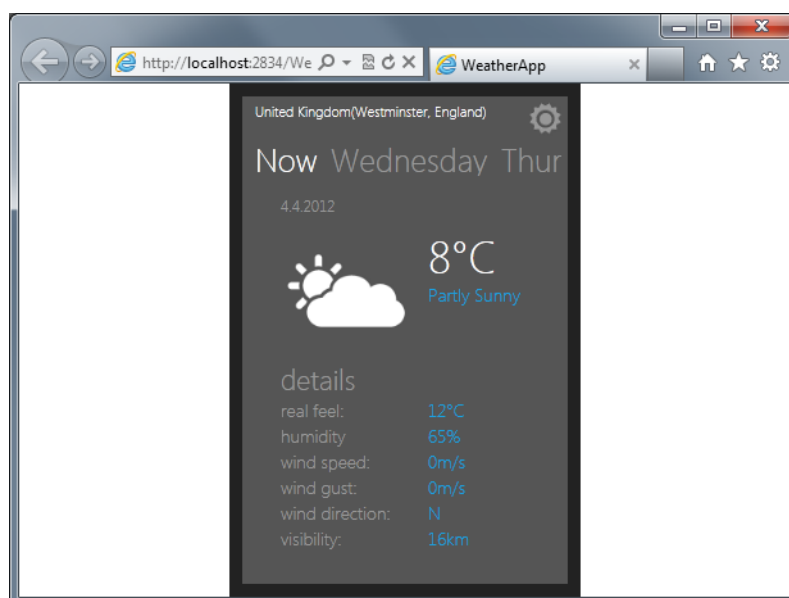
```

Výpis 19: Deklarace parametrů aplikace video přehrávače v HTML definici



### 5.3 Předpověď počasí

Aplikace Předpověď počasí využívá znalostí kapitol kurzu a možnosti režimu „Out of Browser“, tedy může běžet i mimo prohlížeč. Tato aplikace měla za úkol ukázat vlastnosti technologie Silverlight. Aplikace umožňuje vyhledávání měst a následně informací o aktuálním stavu a předpověď počasí nalezené lokace nebo města. Aplikace z počátku používala službu AccuWeather, bohužel použití této služby je pro jakékoli nelicencované použití zakázáno. Z toho důvodu nebude aplikace s touto službou komunikovat, ale bude pracovat s vlastními datovými soubory XML.



Obrázek 20: Aplikace Předpověď počasí

**Požadavky:**

- Metro styl
- Snadné ovládání
- Využití animací
- Stahování a parsování dat XML souboru
- Načítání externích obrázků

Tvorba aplikace byla rozdělena do čtyř fází. První fáze reprezentovala grafický návrh. V druhé fázi se vytvářela struktura datových XML nesoucí informace o nalezených lokacích a o počasí. Ve třetí fázi se tvořila knihovna zpracovávající stažená data XML do objektové struktury. V poslední fázi se realizoval grafický návrh a logika aplikace, využívající vytvořenou knihovnu.

První struktura XML reprezentuje seznam a informace nalezených lokací. Element **citys** zaobaluje seznam elementů **location**. Každý element **location** nese v atributech informace, jako například název města, stát, ve kterém se město nalézá a kód lokace.

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <adc_database xmlns="">
3   < citylist us="10" intl="12" extra_cities="0">
4     <location city="London" state="United Kingdom(Westminster, England)"
5       location="EUR|UK|UK241|LONDON|"/>
6     ...
7   </ citylist >
8 </adc_database>

```

Výpis 20: Struktura XML nalezených lokací

Z důvodu jednoduchosti se provádí rozparsování staženého xml souboru nalezených lokací v řetězci pomocí dotazovacího jazyka „LINQ“ (LINQ to Objects). Samotné parsování probíhá ve dvou fázích. V první fázi se datový řetězec rozparsuje pomocí metody `Parse` třídy `XDocument`. V druhé fázi se provádí selekce lokací, a to tak že se vyberou všechny tagy s názvem „locations“ a následně se vytvoří `List`, jehož každý záznam obsahuje informace o jednotlivých lokacích. Po dokončení parsování se vyvolá událost `WeatherDownloadAndParseCitysComplete`, která vrací získaný `List` s lokacemi.

```

1 public void ParseCitys(string data) {
2     string Data = data.Replace("<?xml version='1.0' encoding='iso-8859-1'?">", "");
3     Data = Data.Replace("<adc_database xmlns='\"'>", "");
4     XDocument x = XDocument.Parse(Data.Trim());
5     List<City> itemList = (from city in x.Descendants("location")
6         select new City{
7             Name = city.Attribute("city").Value,
8             State = city.Attribute("state").Value,
9             Location = city.Attribute("location").Value
10        }).ToList();
11     Citys = itemList;
12     if (WeatherDownloadAndParseCitysComplete != null)
13         WeatherDownloadAndParseCitysComplete(this, new ParseCityArgs(Citys));
14 }

```

Výpis 21: Metoda ParseCitys

Druhá struktura XML reprezentuje informace o konkrétní lokaci, jako například aktuální stav a předpověď počasí na x dnů. Tato struktura datového XML souboru se nalézá v příloze A.1.

Vše co je možné vytvořit v jazyce XAML je možné vytvořit také v jazyce C# nebo VB. Příkladem je animace typu DoubleAnimation, která animuje StackPanel pomocí vlastnosti Canvas.Left. Animace nabývá hodnoty vypočítané pozice StackPanelu a trvá 200ms.

```

1 Duration duration = new Duration(TimeSpan.FromSeconds(0.2));
2 DoubleAnimation myDoubleAnimation1 = new DoubleAnimation();
3 myDoubleAnimation1.Duration = duration;
4 Storyboard sb = new Storyboard();
5 sb.Duration = duration;
6 sb.Children.Add(myDoubleAnimation1);
7 Storyboard.SetTarget(myDoubleAnimation1, DayPanel);
8 Storyboard.SetTargetProperty(myDoubleAnimation1, new PropertyPath("(Canvas.Left)"));
9 myDoubleAnimation1.To = -pos;
10 sb.Begin();

```

Výpis 22: Úryvek těla metody MoveDayPanel

## 6 Závěr

Cílem práce bylo vytvořit rozšíření kurzu týkající se platformy Silverlight 4 a vytvořit sadu screencastů a doprovodných aplikací. Těchto cílů bylo dosaženo a práce byla začleněna do celku kurzu zveřejněného na stránkách <http://silverlight.cs.vsb.cz>. Doprovodné aplikace jsou volně ke stažení na stránkách kurzu a je možné je libovolně používat.

Rozšíření existujícího kurzu touto prací, nemusí znamenat konečnou fázi vývoje kurzu. Do budoucna lze kurz rozšířit o další vlastnosti platformy Silverlight, jako jsou například nové možnosti verze 5, a ukázky využití v operačním systému Windows Mobile 7. Přidáním interaktivní části se kurz jistě stal atraktivnějším a zajisté naučnejším.

Kurz má od svého založení 8276 návštěv z toho 3525 je unikátních návštěvníků. Jeden návštěvník navštíví v průměru 2.48 stránek kurzu. Nejčetnější návštěvy se objevovaly v zimním semestru, kdy se na VŠB vyučuje předmět Vývoj internetových aplikací, kde se mimo jiné vyučuje platforma Silverlight. Je tedy zjevné, že studenti kurz hojně navštěvují a hledají zde inspiraci ke svým projektům.



Obrázek 21: Graf návštěvnosti kurzu za poslední půl rok

Platforma Silverlight otevřela nové možnosti RIA aplikací, ale bohužel se dostala do podvědomí jen málo uživatelům. Dnes se tato platforma nalézá spíše na specializovaných webech a webech společnosti Microsoft a business aplikacích. Možnosti této platformy jsou velké, ale i přes to je budoucnost platformy Silverlight nejistá.

---

## 7 Reference

- [1] Chris Anderson. *Pro Business Applications with Silverlight 4*. Apress, 2010.
- [2] Gill Cleeren. *Microsoft Silverlight 4 Data and Services Cookbook*. Packt Publishing, 2010.
- [3] Derek Franklin. *Flash MX*. CPress, 2003.
- [4] Jaroslav Jirava. *Styly a jejich použití*. <http://www.xaml.cz/wpf/styly-a-jejich-pouziti/>, 2010.
- [5] Luboslav Lacko. *Silverlight*. Cpress, 2010.
- [6] Matthew MacDonald. *Pro Silverlight 4 in C#*. Apress, 2010.
- [7] Microsoft. Msdn library. [http://www.msdn.microsoft.com/en-us/library/cc838158\(v=vs.95\).aspx](http://www.msdn.microsoft.com/en-us/library/cc838158(v=vs.95).aspx).
- [8] Root.cz. Nejsou RIA aplikace cestou do pekla? <http://www.balwan.blog.root.cz/2009/01/29/ahoj-vsichni/>.

## A Přílohy dokumentu

### A.1 Datový XML soubor počasí

```

1 <?xml version="1.0" ?>
2 <adc_database xmlns="">
3   <units>
4     <temp>C</temp>
5     <dist>km</dist>
6     <speed>m/s</speed>
7     <pres>mb</pres>
8     <prec>cm</prec>
9   </units>
10  <local>
11    <city>London</city>
12    <state>Westminster, England United Kingdom</state>
13  </local>
14  <currentconditions>
15    <temperature>7</temperature>
16    <realfeel>4</realfeel>
17    <humidity>87%</humidity>
18    <weathertext>Mostly Cloudy</weathertext>
19    <weathericon>06</weathericon>
20    <windgusts>3</windgusts>
21    <windspeed>3</windspeed>
22    <winddirection>E</winddirection>
23    <visibility>4</visibility>
24    <precip>0.0</precip>
25    <uvindex index="1">Low</uvindex>
26  </currentconditions>
27  <forecast>
28    <day number="1">
29      <obsdate>3/2/2012</obsdate>
30      <daycode>Friday</daycode>
31      <sunrise>6:40</sunrise>
32      <sunset>17:45</sunset>
33      <daytime>
34        <txtshort>A blend of sun and clouds</txtshort>
35        <txtlong>A blend of sun and clouds</txtlong>
36        <weathericon>04</weathericon>
37        <hightemperature>12</hightemperature>
38        <lowtemperature>7</lowtemperature>
39        <realfeelhigh>9</realfeelhigh>
40        <realfeellow>7</realfeellow>
41        <windspeed>3</windspeed>
42        <winddirection>E</winddirection>
43        <windgust>4</windgust>
44        <maxuv>1</maxuv>
45      </daytime>
46      <nighttime>
47        <txtshort>Mostly cloudy</txtshort>
48        <txtlong>Mostly cloudy</txtlong>
49        <weathericon>38</weathericon>

```

```
50      <hightemperature>12</hightemperature>
51      <lowtemperature>7</lowtemperature>
52      <realfeelhigh>9</realfeelhigh>
53      <realfeellow>7</realfeellow>
54      <windspeed>2</windspeed>
55      <winddirection>SSE</winddirection>
56      <windgust>4</windgust>
57      <maxuv></maxuv>
58    </nighttime>
59  </day>
60 </forecast>
61 </adc_database>
```

Výpis 23: Struktura XML detailu nalezené lokace

## **B Přílohy**

1. Webové stránky interaktivního kurzu
2. CD s kompletním obsahem práce
  - Webové stránky interaktivního kurzu
  - Zdrojové kódy projektů
  - Screencasty